

[MICRO]SERVICES FOR FRONTEND

Eugene Naumenko/May 28, 2020

CONTEXT

- Web UIs had won
- UIs evolve, need to mash up, extend scope
- UI programming is the same as any other programming
(backend ~= frontend if requests ~= clicks)
- Technology is an enabler, not a liability
- SOA vs microservices

RATIONALE

- Business needs spectrum:
 - variety of applications: from augmented pages to SPAs to mash-ups
 - faster development, shorter TTM
 - UI/UX as a higher-order entity
- Engineering needs spectrum: all the NFRs
- Management needs spectrum: development process, collaboration, ownership

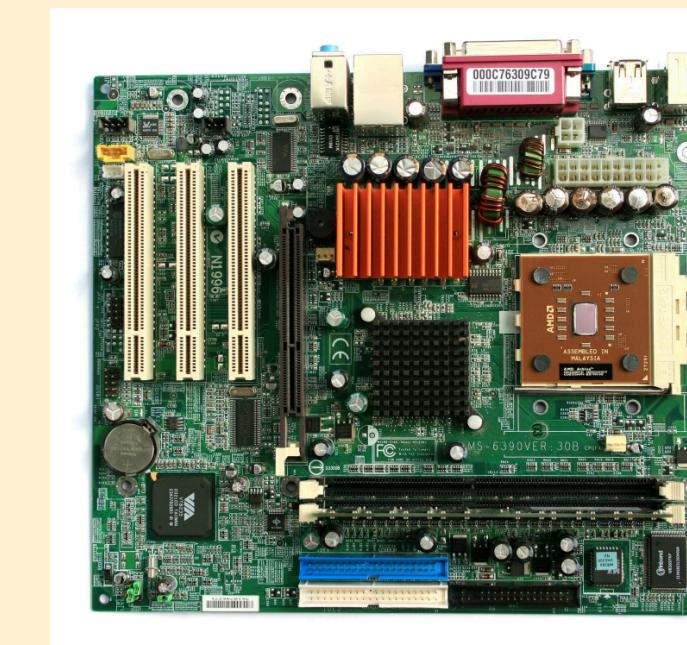
KEY IDEAS

- Composable stateful components
- Composition axis
- Interaction
- Orchestration
- Initialization
- Linking
- Internal structure
- True Reuse (architectural algebra & coalgebra)
- State machines all the way down
- Reactive (N+1 dimensions)
- Programmatic/DSL building
- Declarative bindings (HTML as DSL)
- State management

KEY IDEAS: LOCALLY STATEFUL

Why state encapsulation?

```
t ::=  
  x  
  λx.t  
  t t
```



$$a^1 = a \equiv () \rightarrow a \equiv a \Leftrightarrow \text{“} \text{having a value and having a (pure) program that computes that value.} \text{”}$$

- State management
- State synchronization
- State versioning
- State replay
- State testing
- CRDTs

KEY IDEAS: COMPOSITION IN CODE

- UI structure vs Architecture
- UI-less components
- Stateless components
- UI as a return value
- A whole app could be a component in another app

KEY IDEAS: UI STREAMS

Passing components a UI handle to draw into

vs

Returning a stream of UI updates

KEY IDEAS: STREAMING INTERFACES

- Reactive relations
- N+1 dimensions, Lagrangians

KEY IDEAS: ORCHESTRATION

- Static binding
- Late binding
- Message Broker
- Service Registry
- Service Locator
- Service Bus
- Service Mesh
- Inter-Service Communication Protocol
- Dynamic dispatch
- Singletons

KEY IDEAS: INITIALIZATION

- Bootstrap
- Environment
- Configurations
- Global, shared resource handles
- more

KEY IDEAS: LINKING

- Static (sync)
- Dynamic sync
- Dynamic async

KEY IDEAS: INTERNAL ARCHITECTURE

- Doesn't matter much
- Many different ones could be used
- MVC/Elm arch/etc
- Many frameworks could be used
- Common widgets library and IO primitives are very helpful

KEY IDEAS: UI MASHUPS

- Enterprise scope
- Programmatic UI composition
- Different origins/iframes
- Standard interfaces

KEY IDEAS: UI ALGEBRAS&COALGEBRAS

- A DSL for application creation
- Based on high-level business processes and flows
- Built visually and/or by non-programmers

ARCHITECTURE VALIDATION

- Variant 1:
 - Had been built in 2012
 - Still is being used in production
 - Had been able to embrace React (sic!)
 - Had been able to embrace 10+ programming languages
- Variant 2:
 - Pure functional/Haskell, FRP based
 - 3 days Time To Market

(Q→A) → Zoom Q → Zoom A